

The Finnish Proposition Bank

**Katri Haverinen¹ · Jenna Kanerva¹ · Samuel Kohonen¹ ·
Anna Missilä¹ · Stina Ojala¹ · Timo Viljanen¹ ·
Veronika Laippala² · Filip Ginter¹**

Published online: 16 September 2015

© The Author(s) 2015. This article is published with open access at Springerlink.com

Abstract We present the Finnish PropBank, a resource for semantic role labeling (SRL) of Finnish based on the Turku Dependency Treebank whose syntax is annotated in the well-known Stanford Dependency (SD) scheme. The contribution of this paper consists of the lexicon of the verbs and their arguments present in the treebank, as well as the predicate-argument annotation of all verb occurrences in the treebank text. We demonstrate that the annotation is of high quality, that the SD scheme is highly compatible with PropBank annotation, and further that the additional dependencies present in the Turku Dependency Treebank are clearly beneficial for PropBank annotation. Further, we also use the PropBank to provide a strong baseline for automated Finnish

✉ Filip Ginter
ginter@cs.utu.fi

Katri Haverinen
kahaveri@gmail.com

Jenna Kanerva
jmnybl@utu.fi

Samuel Kohonen
samuel.kohonen@utu.fi

Anna Missilä
anna.missila@utu.fi

Stina Ojala
stina.ojala@utu.fi

Timo Viljanen
timo.viljanen@utu.fi

Veronika Laippala
mavela@utu.fi

¹ Department of Information Technology, University of Turku, 20014 Turku, Finland

² Department of French Studies, University of Turku, 20014 Turku, Finland

SRL using a machine learning SRL system developed for the SemEval'14 shared task on broad-coverage semantic dependency parsing. The PropBank as well as the SRL system are available under a free license at <http://bionlp.utu.fi/>.

Keywords PropBank · Finnish · SRL

1 Introduction

While syntactic parsing reveals much useful information about a sentence, it is still a long way from providing a full analysis of the meaning. As Palmer et al. (2010) expressed the issue, it does not tell “Who did What to Whom, and How and When and Where?” This information is the target of *semantic role labeling* (SRL), the automatic analysis of predicate argument structures. SRL is an important step in grasping sentence semantics, and it has applications in for instance machine translation, information extraction and question answering.

For this reason, work on semantic role labeling is seen as an important task in natural language processing. This is demonstrated by for instance the fact that SRL has been the target of four CoNLL shared tasks: first twice as a stand-alone task (Carreras and Màrques 2004; Carreras and Màrquez 2005), and later twice combined with dependency parsing (Surdeanu et al. 2008; Hajič et al. 2009). SRL is also related to *semantic parsing*, as exemplified by the SemEval'14 Shared Task on broad-coverage semantic dependency parsing.¹

Like many other NLP tasks, the state of the art in SRL uses statistical methods to learn the roles from a previously annotated corpus, which naturally requires that such a corpus be available for the language under consideration. Especially English has seen a large amount of work on SRL, but the necessary resources are available also for other languages (see Sect. 2). This is, however, not the case for Finnish, which has only recently gained a treebank, and for which no SRL resources are previously available. The purpose of this work is to remedy this problem, which has prevented research on statistical SRL for Finnish.

In this work, we present and make freely available the *Proposition Bank* or *PropBank* for the Finnish language, constructed on top of the previously existing Turku Dependency Treebank (TDT) (Haverinen et al. 2013b).² This new resource enables the study of Finnish semantic role labeling, which has previously been impossible, and thus opens a path towards advanced semantic applications for this language.

2 Related work

As pointed out by Palmer et al. (2010), the task of defining a universal set of semantic roles has proven difficult. This can perhaps be seen from the fact that there have been several independent efforts targeting the semantic roles of English. The

¹ <http://alt.qcri.org/semeval2014/task8/>.

² Parts of this work have previously been described in a conference paper by Haverinen et al. (2013a).

three best known ones are undoubtedly the FrameNet (Baker et al. 1998), VerbNet (Dang et al. 1998) and PropBank (Palmer et al. 2005) projects.

Out of these three resources, FrameNet uses the most fine-grained labels. For instance, to verbs of cooking, FrameNet assigns roles such as *food* and *cook*. VerbNet is similar in the sense that it defines roles for groups of verbs at a time, but it uses more generic labels, such as *agent* and *patient*. PropBank is the most coarse-grained of the three, as it uses numbered labels for its roles and defines these roles on a verb-by-verb basis. For a more detailed description of the PropBank scheme, see Sect. 3. The purposes of the three resources are also slightly different. Broadly stated, FrameNet and VerbNet are intended to be lexicons of verbs, whereas PropBank is the only one of the three intended as a corpus of semantic roles annotated in running text. However, as shown for instance by Giuglea and Moschitti (2004, 2006), there is a clear relation between FrameNet on one hand and PropBank and VerbNet on the other. There is also a close connection between PropBank and VerbNet as we will discuss in Sect. 3.

For languages other than English, especially the PropBank annotation scheme has been popular. Several other PropBanks have emerged after the original work, including among others Chinese (Xue and Palmer 2009), Arabic (Zaghouani et al. 2010), Hindi (Palmer et al. 2009) and Brazilian Portuguese (Duran and Aluísio 2011).

As for SRL systems, the seminal work was that of Gildea and Jurafsky (2002). Later on, studies by for instance Toutanova et al. (2008), Pradhan et al. (2008), Johansson and Nugues (2008), Punyakanok et al. (2008), Surdeanu et al. (2007) and Moschitti et al. (2008) have investigated the details of the topic further. Also the CoNLL shared tasks of 2004 and 2005 (Carreras and Màrques 2004; Carreras and Màrquez 2005) have resulted in a large number of systems for English. In the biomedical domain, the system of Barnickel et al. (2009), SENNA, is targeted at efficient extraction of semantic role labels at a large scale.

The only existing semantic role labeling resource for Finnish is a small-scale clinical Finnish PropBank by Haverinen et al. (2010). This is a pilot study, part of whose purpose was to establish the feasibility of Finnish PropBanking, using a limited amount of data from the narrow domain of patient reports in an intensive care unit.

3 Proposition Banks and terminology

A PropBank consists of two parts: a lexicon of verbs and an annotated corpus that uses the definitions in it. Each verb in the lexicon is given a number of *framesets*, which correspond to coarse-grained senses of the verb. Each frameset contains a number of *roles* or *arguments* for its verb sense, thus defining semantic roles on a verb-by-verb basis. The arguments are numbered from zero onward, resulting in labels, e.g., *arg0* and *arg3*, and given a free-text description. The argument labels zero and one are special, reserved labels: *arg0* is intended for *agents*, *causers* and *experiencers* and *arg1* for *patients* and *themes*. Arguments beginning from *arg2* have no predefined meanings, but rather they are defined separately for each verb.

However, the original PropBank makes an effort to keep argument numberings consistent for verbs within the same VerbNet (Dang et al. 1998) class, and the Finnish PropBank strives for a similar consistency goal using the *batch* system of frameset assignment described in Sect. 6.3. Table 1 illustrates the concept of framesets.

Outside the framesets, PropBank defines a set of 13 general-purpose *adjunct-like arguments* or *ArgMs*, with labels such as *CAU* (*cause*) or *DIR* (*direction*). The Finnish PropBank defines two additional ArgMs: *CSQ* (*consequence*) and *PRT* (*phrasal marker*). Unlike the numbered arguments, ArgMs can occur together with any verb. The distinction of numbered and adjunct-like arguments is based on frequency: if an argument candidate frequently occurs with a verb sense, it is defined as a numbered argument in the frameset, and otherwise it is made an adjunct-like argument.

4 Corpus: The Turku Dependency Treebank

One of the properties of a PropBank is that it is annotated on top of a treebank and tightly bound with it; in the case of the English PropBank, this is the Penn Treebank (Marcus et al. 1993). For the PropBank, this means that the arguments cannot be simply any string of words, but are restricted by the underlying treebank. In the English PropBank, arguments are required to be *constituents* of the Penn Treebank (or in some cases, combinations thereof).

The PropBank presented in this paper is built on top of the *Turku Dependency Treebank* (TDT) (Haverinen et al. 2013b). This treebank consists of 204,399 tokens (15,126 sentences) of text from 10 different genres of general Finnish, such as the Finnish Wikipedia, financial news and amateur fiction.

The syntactic analyses of the treebank have been annotated manually using a Finnish-specific version of the well-known Stanford Dependency (SD) scheme (de Marneffe and Manning 2008a, b; de Marneffe et al. 2013). It consists of two layers: the *base* layer, where the analyses are expected to be trees, and the *conjunct propagation and additional dependencies* layer, which adds further dependencies on top of the *base* layer analyses, thus making them graphs rather than trees. This is in order to give more information on phenomena that could not be fully analyzed in the *base* layer due to the treeness restriction.

Of the phenomena analyzed in the second annotation layer, three are relevant to the PropBank annotation:

<i>Propagation of conjunct dependencies</i>	As in the SD scheme the first element of a coordination is marked as the head, it is not possible, in the <i>base layer</i> , to distinguish modifiers of the head from modifiers of all (or some) conjuncts. Thus this distinction is made in the second layer by <i>propagating</i> dependencies to all conjuncts that they relate to.
---	--

<i>External subjects</i>	If two verbs share a subject in subject control, only one of the subjects can be marked in the <i>base layer</i> , and the other must be marked as an <i>external subject</i> in the second layer.
<i>Syntactic functions of relativizers</i>	The phrases with the relative word are marked only as <i>relativizers</i> in the <i>base layer</i> tree. Their secondary syntactic function, which can in principle be any function defined by the <i>base layer</i> , is thus marked in the second layer.

The second annotation layer was added in part for the purposes of creating the PropBank, and indeed, it turned out to be important for the compatibility of SD and the PropBank scheme, as will be discussed in Sect. 8.

In addition to the syntactic annotation, the treebank contains morphological analyses based on the output of OMorFi (Pirinen 2008; Lindén et al. 2009), an open source tool for Finnish morphology. For each token, OMorFi gives its all possible readings, of which one is subsequently selected using a machine learning method. However, since this machine learning based selection is not manually corrected, we chose to not utilize this disambiguation step and use for every word the full set of its morphological analyzes. Thus, if a token *can* be analyzed as a verb, it is selected for annotation irrespective of whether the verbal reading is selected in the treebank. While this strategy increases the workload, it ensures higher recall. As will be described in more detail in Sect. 6.2, those tokens that are not in fact verbs are marked as such during the annotation.

The syntactic and morphological analyses of the treebank are illustrated in Fig. 1. For further details on the treebank, we refer the reader to the paper by Haverinen et al. (2013b) and the annotation manual by Haverinen (2012).

5 PropBanking and the Stanford Dependency scheme

The English PropBank is built on top of the constituency-based PennTreebank (Marcus et al. 1993), and accordingly, it associates its arguments with the constituents of the treebank. In principle, any constituent can be an argument. The Finnish PropBank, however, is built on top of the Stanford Dependency scheme, and therefore our approach to PropBanking is necessarily somewhat different.

In this work, we associate arguments with *dependencies* from both annotation layers of the underlying treebank. This does not, however, mean that we assume all arguments to be direct syntactic dependents of the verb. Whenever an argument is found outside the verb's dependents, the annotator is to add a new dependency of the type *xarg* (*external argument*) during PropBank annotation, and the argument is associated with this dependency. Figure 2 illustrates the annotation of the PropBank and the external arguments. Typical cases of external arguments include for instance structures with participial (see Fig. 2) or infinitival modifiers.

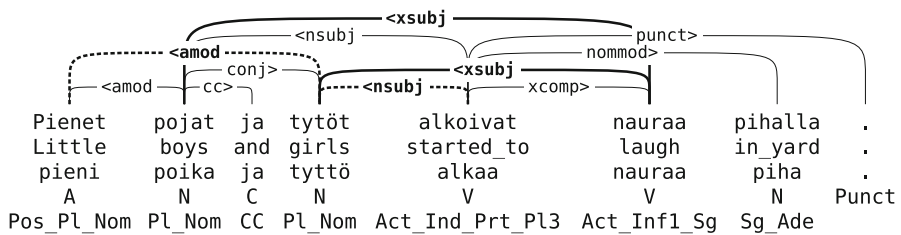


Fig. 1 The syntactic and morphological analyses in TDT. The syntactic analyses are represented as a graph of dependencies. The *base* layer dependencies form a tree structure, and the *conjunct propagation and additional dependencies* layer is added on top of the tree (**bold dependencies**). The dependencies marking conjunct propagation are *dashed*. The syntactic structure reveals for instance that both *pojat* (*boys*) and *tytöt* (*girls*) act as the subject to the verb *alkoivat* (*started*). The morphological analyses of the words are marked under each word in the figure. For instance, the word *pojat* (*boys*) is known to be a form of the lemma *poika* and as such a noun in the plural nominative form. The example can be translated as *The little boys and girls started to laugh in the yard*

This approach to PropBanking has a feature worth noting. In most cases, the argument can be interpreted as the dependent word and its full syntactic subtree, where *subtree* is defined by the *base* layer of syntactic dependencies. However, as shown by Choi and Palmer (2010), not all arguments associated with dependencies necessarily have the correct boundaries with this interpretation. In their work, Choi and Palmer convert the original PropBank into a dependency format, by first automatically converting the underlying Penn Treebank into a dependency scheme, and subsequently retrieving the arguments from the PropBank. They then list a number of cases in which this strategy results in incorrect argument boundaries.

However, the majority of these cases do not apply for the Finnish PropBank, for two main reasons. First, the underlying treebank, TDT, is annotated using the SD scheme, which, although a syntax representation, is semantically motivated and thus steers clear of several of these issues. For instance, in the dependency scheme used by Choi and Palmer, the auxiliary acts as the syntactic head of its main verb. This is in conflict with a PropBank, where the auxiliary should become an *ArgM-MOD* for its main verb. In contrast, the SD scheme makes the auxiliary a dependent of the main verb, which is as desired for PropBanking purposes. Similarly, in the syntax scheme used by Choi and Palmer, a negation can, in some coordination structures, become the head of its main verb, and in the PropBank scheme, the negation should be an *ArgM-NEG* for the main verb. Again, this is compatible with the SD scheme, as it always makes the negation depend on its main verb regardless of any coordinations possibly present.

Second, TDT is natively annotated in the SD scheme, rather than converted from a constituency scheme, and an emphasis on attachment issues was present in the annotation work. Therefore, TDT may be more likely to agree with the PropBank annotation. Choi and Palmer show an example where the dependency conversion results in two arguments overlapping each other in the dependency annotation, because one of them becomes a dependent of the other, rather than a direct dependent of the verb. In contrast, in the native annotation of TDT, a similar

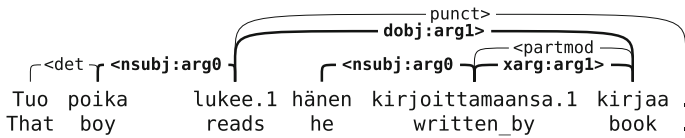


Fig. 2 The dependency-based PropBank annotation. Arguments are associated with dependencies, and for arguments outside the direct dependents of the verb, a new dependency of the type *xarg* is added. The *base* layer syntax subtree defines the argument, so that for instance the *arg0* argument for *lukee* (*reads, is reading*) is *tuo poika* (*that boy*). The example can be translated as *That boy is reading a book written by him*

situation would be annotated in such a way that both arguments would be dependents of the verb to begin with.

This is not to say that none of the cases listed apply to TDT. In fact, one such case is present in Fig. 2. Consider the argument 1 for the verb *kirjoittamaansa* (*written by*). From the assumption that arguments are full subtrees of the dependent word, it would follow that the argument 1 of *kirjoittamaansa* would include not only the noun *kirjaa* (*book*) (as intended) but also the verb *kirjoittamaansa* itself, which is naturally incorrect. However, we expect this phenomenon to be considerably rarer than in a dependency conversion of a constituency treebank, and additionally, most cases are likely simple to solve. In the case present in Fig. 2, for instance, it suffices to forbid the argument to include the verb itself. Naturally, if one were to use this resource for an application that requires knowing the exact spans of each of the arguments (Choi and Palmer mention machine translation as an example), these remaining cases would need resolving.

6 Framing and annotation

In this section, we discuss the details of framing and annotation. We begin with the workflow of the annotation and then move on to briefly describing the annotation software. Finally, we discuss the *batch* system of creating framesets and the resulting connections between framesets, which are to our knowledge unique to the Finnish PropBank.

6.1 Annotation workflow

As a PropBank consists of two parts—the framesets and the annotated corpus—its construction workflow consists of two phases as well. In the first phase, termed *framing*, a verb is given its framesets based on the occurrences present in the text corpus. This means that only framesets needed for the annotation are created. In the second phase, then, these framesets are used in annotating the occurrences.

A total of six annotators with differing backgrounds took part in the creation of the PropBank, and the same annotators also acted as framers. We used a workflow combining single and double annotation, so as to optimize the speed on one hand, and quality on the other.

Table 1 Two framesets for the English verb *to lift*

lift.01	<i>Cause increase in elevation</i>	lift.02	<i>Acquire stealthily, remove</i>
arg0	Elevator, agent	arg0	Agent, entity acquiring/removing something
arg1	Logical subject, patient, thing rising	arg1	Thing acquired/removed
arg2	EXT, amount risen	arg2	Acquired/removed-from
arg3	Start point		
arg4	End point		

The frameset *lift.01* is meant for usages such as *John lifted a heavy box onto the table*, and *lift.02* for cases like *John lifted Mary's keys from her purse*. The *EXT* in the first frameset refers to ArgM-EXT (extent), one of the adjunct-like argument types defined outside of framesets

Table 2 A frameset for the English verb *to move* in the sense *to change the location of something*

move.01	<i>Change location</i>
arg0	Mover
arg1	Moved
arg2	Destination

Only one frameset is required for three different uses of the verb: this frameset can appear with either an arg0 or an arg1, or with both

The overall annotation protocol was as follows. Verb lemmas were identified using OMorFi (Pirinen 2008; Lindén et al. 2009), which gives each word all of its possible readings, meaning that all tokens that could possibly be verbs were considered for annotation. This was to make sure that all verb tokens would receive a PropBank analysis, even in case of errors in the automatically assigned readings of the treebank. Each verb lemma was assigned to either one or two annotators. First the annotators were to create a preliminary set of framesets for the verb based on reviewing a sample of the occurrences. If there were two annotators, they were to either construct the framesets together or to divide the work so that one annotator constructed the frameset and the other inspected and approved it. For single annotated verbs, the framesets were created by the annotator alone, but for each verb another annotator was assigned as a *consultant*, with whom the annotator could discuss if the framing was problematic.

After the framing, the occurrences were to be annotated according to the framesets created, and in double annotation, annotators were to annotate the occurrences independently of each other. However, if an annotator felt that either a frameset was unsuitable for its purpose or some occurrence was unaccounted for by the current framesets, it was possible to discuss and modify the framesets, and even create new ones, in mid-annotation.

In double annotation, when both annotators were finished annotating all occurrences assigned to them, the annotations were automatically *merged* together so that both options could be seen whenever a disagreement occurred. A meeting with the whole annotation team was then held to discuss these disagreements and

Table 3 A frameset for the Finnish intransitive verb *liikkua*

liikkua	<i>Move</i>
arg0	Entity moving
arg1	Entity or thing moving If not arg0
arg2	Destination

The frameset has both an arg0 and an arg1, but they are mutually exclusive, meaning that only one of them should be annotated in any given occurrence of *liikkua*

Table 4 A frameset for the Finnish transitive verb *liikuttaa* (to make something move)

liikuttaa	<i>Move something</i>
arg0	Entity moving something
arg1	Entity or thing moving
arg2	Destination

The frameset resembles the English frameset for *to move*, but whereas the English verb can appear with either arg0 or arg1 in the subject position, *liikuttaa* is transitive and must have a subject arg0

decide on a single, best analysis. In the same context, cases marked as *unsure* by annotators (in either double or single annotation) were also discussed and settled.

The project was started using full double annotation to be able to set rules for possible difficult cases and to ensure good annotation quality even in the learning phases of the work. Later, as preliminary quality evaluations showed promising results, the work was steered towards single annotation to increase the speed. In this setting, verbs with a large amount of occurrences were partially double annotated, meaning that some percentage of their occurrences were double annotated and the rest were single annotated. Verbs with only few occurrences were completely single annotated.

6.2 Annotation software

The annotation was done using a custom software with two parts. The first part, the *frameset editor*, allows an annotator to create and edit framesets for verbs. Each verb can be given any number of framesets, and each frameset consists of a definition, a free comment field and a number of arguments. Each argument, in turn, consists of a definition and a free comment field. The comment fields are used for, for instance, usage notes and case restrictions.

The second part of the software is the *annotation tool*. It finds all verb lemmas assigned to a certain annotator and displays them as a list. The annotator can then select a lemma to work on. The tool will display each occurrence as a separate case, and one of the following actions must be taken. First, it is possible to select one of the framesets created for this verb and annotate the arguments accordingly, then mark the occurrence as *set* when all arguments have been marked. Second, if the

Table 5 Framesets for the verbs *pukea*, *to dress* (left) and *pukeutua*, *to dress oneself* (right)

pukea	<i>Dress</i>	pukeutua	<i>Dress oneself</i>
arg0	Person who dresses	arg0	Person dressing oneself
arg1	Person wearing clothes		
arg2	Clothes	arg2	Clothes

The framesets are identical otherwise, but because with *pukeutua* arg0 is always dressing arg0, its frameset has no arg1. The frameset for the transitive verb is identical to that of the English verb *to dress*

occurrence under consideration is not a verb, the annotator may mark it as such using the *not a verb*-function. Third, if the occurrence is a verb, but the lemma suggested is incorrect for it, the annotator is to mark it as *wrong lemma*. Finally, as auxiliaries are not given framesets or arguments in a PropBank, the *auxiliary* function may be used.

In addition, if an annotator feels uncertain about a frameset or argument, any argument may be marked as *unsure*. The same function can be used to mark suspected syntax-level errors, as annotators are not allowed to alter the syntax trees at this stage. These markings are then discussed in the team meetings, as described in Sect. 6.1.

6.3 Re-using framesets and resulting connections

In addition to creating a frameset from scratch, the annotators had two further options, which served to decrease the work-intensity of the frameset creation as well as to enforce some consistency within the PropBank. Similar consistency was sought for in the English PropBank by striving to assign argument numbers consistently within a VerbNet class.

First, if an annotator believed that the frameset currently under construction would fit other verbs, it was possible to use a *batch system* to assign the frameset to multiple verb lemmas at once. For instance, consider verbs of affection. When constructing a frameset for the verb *to like*, it would be useful to be able to give the same frameset to other verbs with a similar meaning and the same argument structure, such as *to love*, *to adore*, *to care* and *to dig*. This is the purpose that the batch system was designed for: creating a single frameset and assigning it to multiple verbs at once. The annotators were instructed to batch-assign framesets to verbs that share the same PropBank arguments, including argument descriptions. Therefore, in our verbs of affection example, the verbs listed would be part of the same batch, whereas verbs of dislike, which have the same numbered arguments but unsuitable argument descriptions, would not. It was also possible to modify the framesets once created, even if they were originally part of a batch.

As a minor downside of this approach, it should be noted that the different verbs in a batch may be assigned to different annotators, thus necessitating further coordination. Care must be taken that no new framesets are given to verbs without the knowledge of their annotators, as otherwise these verbs may receive (near)

Table 6 Framesets for the verbs *sulkea*, to close something (left) and *sulkeutua*, to close by itself (right)

sulkea	<i>Close something</i>	sulkeutua	<i>Close (by itself)</i>
arg0	Entity closing arg1		
arg1	Thing closing	arg1	Thing closing

The verb *sulkea* is transitive and thus has both arg0 and arg1, whereas *sulkeutua*, a reflexive derivation of *sulkea* is intransitive and only takes a patient argument. In the case of the verb *sulkea*, a reflexive derivation with an agent closing him or herself would not be reasonable, and thus the meaning of the reflexive derivation is automative

identical framesets from different sources, or framesets not in fact needed for the verb in question. If necessary, it was also possible to delete superfluous framesets.

The second way for annotators to use the same frameset for multiple verbs was to copy existing framesets to new verbs. When creating a frameset, the annotators could view the existing framesets of other verbs, and if a suitable one was found, it could be copied for the new verb under consideration. As with the batch system, it was possible to further modify a frameset after it had been copied, and when copying without modifying, similar instructions applied: framesets should only be copied if the free-text descriptions were suitable.

In addition to making the framing process faster and enforcing consistency between similar verbs, the batch system and the frameset copying mechanism have a further, positive consequence. Constructing framesets in batches and copying them from one verb to another causes a link to be created between all verbs involved, meaning that it is possible to find the connection between these verbs also afterwards.

Since these linked framesets connect together verbs with similar meaning and argument structure, these connections can be used among others as a fallback method to collect training data for rare and unseen predicates during semantic role labeling. For example when arguments are predicted in a predicate-wise fashion (i.e. the argument prediction system is trained separately for each predicate), rare framesets have very little training data and framesets occurring only in the test section do not have training data at all. However, these links can be used to merge together rare framesets to get more training data for rare and unseen predicates. Such an approach has been used by Kanerva and Ginter (2014) in their SRL system.

7 Finnish-specific issues

The Finnish PropBank has been constructed using the English PropBank as a reference, and often the framesets for the Finnish PropBank are given the same structure as in the English PropBank. This, in addition to the *batch* system and copying mechanism described in the previous section, serves the purpose of enforcing consistency across the framesets of different verbs. However, it is not always possible to follow the structure of the English framesets.

In this section, we discuss certain regular cases where Finnish verbs behave differently from English verbs, in a way that causes differences between framesets. We begin by examining *causative derivations*, where two Finnish verbs are needed to express the same meanings as one polysemous English verb. We then turn to *reflexive derivations*, which often cause otherwise identical framesets between Finnish and English to differ with respect to either arg0 or arg1. Finally, we briefly discuss differences of framesets in general.

7.1 Causative derivations

Certain English verbs, often termed *variable behavior verbs* (see for instance the work of Levin and Hovav (1994) and Perlmutter (1978)), are polysemous in a systematic way. They are often movement verbs, and indeed, a typical example is the verb *to move*. It can be used in three different ways: with an agent subject (*he moves*), with a patient subject (*the box moves*) or with an agent subject and a patient object (*he moves the box*).

In Finnish, some verbs exhibit the same behavior, but this is not typical. Instead, the same meanings are expressed using two different verbs. For instance, the Finnish verb *liikkua* (*to move*) is intransitive. Thus it can be used with an agent subject (*hän liikkuu*, *he moves*) or a patient subject (*laatikko liikkuu*, *the box moves*), but not transitively with an agent subject and a patient object. For the transitive use, Finnish has a separate verb, *liikuttaa* (literally, *to make something move*), which is a so called *causative derivation* of *liikkua* and can only be used transitively.

This results in differences in the PropBank framesets of Finnish and English. For the English verb *to move* the frameset, as illustrated in Table 2, is rather simple. Both arg0 and arg1 are present in the frameset, and each occurrence is to be annotated with those arguments present in it, as the PropBank scheme does not require all arguments to be present in all occurrences. For Finnish, however, the situation is slightly more complex.

For the intransitive verb *liikkua* we give one frameset, with both arg0 and arg1 and an explicit mention that these two arguments are mutually exclusive and only one of them should be annotated in any given occurrence. See Table 3 for an illustration. The transitive verb *liikuttaa*, in turn, is given a frameset closely resembling the English one for *to move*, with both arg0 and arg1 present (see Table 4 and for comparison, Table 2). However, the argument structures of the Finnish and English verbs are not considered identical, as the English verb can be used either with the agent or the patient as the subject, but in Finnish, only the agent can be the subject.

7.2 Reflexive derivations

Also other somewhat regular differences between the Finnish and English PropBanks exist. Typical cases are those where the English frameset contains an arg0 or an arg1 which is not present for Finnish. Both of these types of differences

can be caused by *reflexive derivations* (see the Finnish Grammar (Hakulinen et al. 2004, §334–335), or for instance the work of Paulsen (2011)).

Reflexive derivations are typically used in two cases. First, they can express a situation where an agent performs an action on him- or herself. For instance, consider the transitive verb *pukea* (*to dress*). Using the affix *-utu*, it is possible to derive *pukeutua*, which has a reflexive meaning: *to get dressed* or *to dress oneself*. With some reflexive derivations, as with *pukeutua*, it is also possible to express the same meaning using the root verb and the reflexive pronoun *itse* (*oneself*). Thus *pukea itsensä* has the same meaning as the reflexive derivation *pukeutua*.

In PropBank terms, this means that the verb *pukeutua* receives a frameset resembling that of the English verb *to dress* (and that of the root verb *pukea*), except that it does not contain a separate arg1, simply because with this verb, arg0 is always dressing arg0. For an illustration of this use of the reflexive derivation and the relevant framesets, see Table 5.

The second use of the reflexive derivation is one where the meaning is *automotive*,³ that is, where something happens by itself. For example, from the verb *sulkea* (*to close*), one can derive *sulkeutua*, which is intransitive and used in cases where something closes by itself or without a known agent—*ovi sulkeutuu*, *the door closes*. For *sulkeutua*, the frameset should resemble one of *to close* and at that, the transitive root verb *sulkea*. The difference is that *sulkeutua* does not take an agent and thus the frameset should lack arg0. Table 6 illustrates the PropBank framesets for an automotive use of the reflexive derivation.

The automotive use is, according to the Finnish Grammar (Hakulinen et al. 2004, §335), often such that an agent performing a reflexive action would be either impossible or not meaningful. It also notes that some verbs, even if the meaning of the verb would be suitable for reflexivization, do not allow the reflexive derivation at all. In such cases it is still possible to express the same meaning using the reflexive pronoun *itse*. For instance, the verb *moittia* (*to scold*, *to criticize*) does not allow a reflexive derivation (**moittitua*), but it is still possible to say *moittia itseään* (*to scold oneself*). The use of a reflexive pronoun is naturally unproblematic for PropBanking, as it serves as the patient (arg1) which would be absent with the reflexive derivation.

It should be noted that the distinction between the reflexive and automotive uses of these derivations is not always clear-cut. According to the Finnish Grammar (Hakulinen et al. 2004, §334), some uses are between the two, and in some cases even the same verb may be used both reflexively and automotively. This is in line with our general observation that even though clear in most cases, the distinction between arg0 and arg1 was one that repeatedly caused clashing intuitions between annotators in the framing phase.

7.3 Other cases

Naturally, not all differences between the two PropBanks are regular. As described in the paper by Haverinen et al. (2013a), there are also cases likely due to contextual

³ This term has been predominately used by fennists, in general linguistics the usual term to be used is *anticausative*. See for instance the work of Comrie (1985).

differences between the underlying treebanks rather than differences between the two languages.

For instance, the Finnish verb *juosta* can be translated as *to run*, but its frameset does not resemble any frameset of *to run*. This is because in TDT, the most common usage of *juosta* describes an agent running from one location to another, but the English PropBank does not contain such a frameset. Instead of the two locations, the English frameset describing an agent running contains an argument for a race, course or distance, which in turn is absent in the Finnish frameset. As it is perfectly conceivable to use the verb *to run* to describe running from a place to another (see for instance the Collins dictionary (2009)), we find it likely that this difference between the PropBanks is due to contextual differences between TDT and the Penn Treebank.

There may also exist differences that are irregular but due to actual differences between the two languages rather than just the contexts of the treebank texts. However, judging whether a particular difference is due to contextual or linguistic differences would not be an easy task, and is out of scope for the Finnish PropBank project.

8 Evaluation

In this section, we present several evaluations of the Finnish PropBank. We begin with a basic evaluation of annotation quality, using measures of *annotator accuracy* and *interannotator agreement*. We then move on to evaluate the compatibility of the SD syntax scheme with the PropBank semantic role labeling scheme, by measuring the coverage of the SD scheme over the PropBank arguments.

Annotation quality can be evaluated using the *annotator accuracy* of our annotators against the annotation present in the final PropBank. This is done using F_1 -score, defined as $F_1 = \frac{2PR}{P+R}$, where P stands for *precision* and R stands for *recall*. Precision is the proportion of annotator arguments that are also present in the gold standard, while recall is the proportion of arguments in the gold standard that are also present in the annotator output.

In order to count as correct, an argument (as viewed through the dependency it is associated with), must have the correct dependent word, and the correct argument label. The head word is the verb and thus always correct. In addition, if the frameset assigned to the verb is incorrect, then all numbered arguments assigned to this occurrence are counted as incorrect as well, seeing that the arguments are defined on a verb-by-verb basis. ArgMs, however, can be counted as correct regardless of the frameset assigned, as they are frameset-independent. Naturally, only the portion of the PropBank that has been double annotated can be evaluated in this way, amounting to 43.0 % of all verb occurrences and 46.1 % of all arguments (Both of these figures are calculated on all *possible* verb tokens, including those that annotators have marked as *not a verb*).

Table 7 presents the main evaluation results. A separate evaluation is given for numbered, adjunct-like and external arguments, as well as an overall evaluation of all of these argument types. The overall *annotator accuracy* across all annotators

and different argument types is 91.7 %, indicating high annotation quality. As also seen from the table, it would seem that the numbered arguments are the easiest to annotate, as compared to the adjunct-like and external arguments. The former result is in line with the results of Palmer et al. (2005), who also reported that adjunct-like arguments were more difficult than numbered ones. The latter is intuitive as well, seeing that external arguments are easy to overlook, and unlike with other arguments, the annotator is required to recognize the correct dependent word for the dependency associated with the argument.

In addition to *annotator accuracy*, we also measure the overall *interannotator agreement* of our annotators, using *Cohen's kappa*. Kappa is defined as $\kappa = \frac{P(A) - P(E)}{1 - P(E)}$, where $P(A)$ is the observed agreement, and $P(E)$ is the agreement expected by chance. Overall, the kappa of our annotators was 85.8 %, which, like the annotator accuracy, indicates high quality.

As an additional point of interest, the frameset assignment can be evaluated separately, simply calculating the percentage of correctly assigned framesets. Out of all frameset assignments in the corpus (twice the number of double annotated occurrences), 97.9 % were correct. Each frameset of the verb under consideration provided one possible choice, and in addition the annotators had the choices *not a verb*, *wrong lemma* and *auxiliary*.

Next, we evaluate the compatibility of the SD and PropBank schemes, by measuring the overlap of syntactic dependencies with PropBank arguments. If we, for the moment, disregard the verb *olla* (*to be*) in our calculations, then out of all PropBank arguments, 81.3 % are syntactic dependents of their verb in the *base* layer of SD. For numbered arguments, this portion is 76.1 %, and for ArgMs, 90.1 %. At this point, the coverage of the SD scheme does not yet seem adequate. However, if we consider both the *base* layer and the *conjunct propagation and additional dependencies* layer, 93.2 % of all arguments are covered—89.7 % of numbered arguments and 99.1 % of ArgMs. This shows the clear benefit provided by the *conjunct propagation and additional dependencies* layer of TDT.

As mentioned above, the verb *olla* was disregarded in the previous calculations. This is because as a copular verb (the only one such in Finnish), it is somewhat of a special case. In the SD scheme, the copula is not marked as the main verb of its clause, but rather a dependent of the predicative, most often a noun or adjective. The rationale for this attachment is semantic, but for PropBanking purposes, it is slightly inconvenient, as the annotation is for arguments of verbs.

For the current purposes, we have annotated the copular *olla* as if it were a regular verb. The relevant frameset has two arguments: *arg1* for the thing that is, and *arg2* for what *arg1* is. Due to the SD attachment of copulas, both of these arguments are regularly *external arguments*, as are any possible adjunct-like arguments they may have. The annotation of copulas is illustrated in Fig. 3. There are in total 3,735 copular verbs in the data, as calculated based on the syntactic construction,⁴ totaling 7.5 % of all verb occurrences, and they result in a total of 10,932 external arguments, which is 71.8 % of all external arguments.

⁴ The same PropBank frameset is used for essive constructions, which have the same arguments but are syntactically different in TDT.

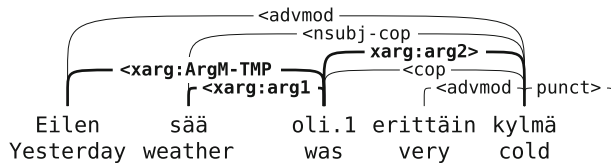


Fig. 3 Annotation of the verb *olla* (to be) as a copular verb. Due to the way that SD attaches copulas, cases of *olla* cause a large amount of external arguments. Note that the annotation must distinguish between adjunct-like arguments of the copular verb [*Eilen* (Yesterday)] and elements genuinely modifying the predicative [(*erittäin* (very))]. The example can be translated as *The weather was very cold yesterday*

9 Semantic Role Labeling baseline

In this section, we establish the baseline performance for Finnish SRL using two separate machine learning systems. First, we have previously used the PropBank to evaluate a novel SRL method combining vector space representations of the lexicon with supervised classification, achieving a labeled F_1 -score of 73.83 % (Kanerva and Ginter 2014). This study focused on the role label assignment task and relied for argument detection almost exclusively on the Finnish syntactic parsing pipeline of Haverinen et al. (2013b) followed by the *conjunct propagation and additional dependencies* layer prediction method of Nyblom et al. (2013). Interestingly, the unlabeled F_1 -score of the method is 89.29 %, using the parser-produced, non-gold syntactic trees. This indicates that the step of identifying the arguments (disregarding role label assignment) is rather successful, confirming that the native annotation in the SD scheme is suitable for SRL when combined with a parser natively trained for the scheme and augmented with the prediction of the conjunct propagation and additional dependencies layer.

Subsequently, we have also trained the semantic role labeling system of Kanerva et al. (2014), which was developed as an entry in the SemEval-2014 Shared Task on broad-coverage semantic dependency parsing. The objective in this SemEval task was to identify and label semantic dependencies on English data in three different annotation schemes. The system ranked third with an average labeled F_1 -score of 80.49 % across the three representations. Its results can thus be considered as a strong, non-trivial baseline.

Several modifications were necessary to account for the differences between the semantic representation and task configuration of the SemEval-14 task for which the system was developed, and the PropBank-based SRL task. First, the SemEval task does not include sense disambiguation (frameset selection) and we therefore extend the system with the word sense disambiguation component from the abovementioned method of Kanerva and Ginter (2014). And second, we restrict the arguments predicted by the system to those where the predicted governor is one of the predicates annotated in the PropBank. This step allows us to draw a direct comparison to the SRL results reported for a number of languages in the CoNLL-09 Shared Task (Hajič et al. 2009). Finally, we extend the system with additional features to address characteristics unique to either Finnish language or the PropBank

Table 7 Annotator accuracy results per annotator and overall

	Ann. 1	Ann. 2	Ann. 3	Ann. 4	Ann. 5	Ann. 6	Overall
Numbered arguments (n = 52,548)							
F	97.4	95.2	93.0	92.7	96.8	91.9	94.9
P	97.9	96.2	94.1	92.9	96.9	91.2	95.3
R	97.0	94.4	91.9	92.6	96.7	92.6	94.6
ArgMs (n = 29,881)							
F	91.2	84.3	84.3	83.2	91.2	78.9	86.1
P	91.3	84.5	83.8	84.7	92.0	78.8	86.2
R	91.2	84.2	84.8	81.8	90.4	79.0	85.9
xargs (n = 5373)							
F	92.6	84.6	78.4	77.4	90.8	80.1	85.6
P	95.9	95.5	87.7	84.1	94.0	81.8	91.0
R	89.5	75.9	71.0	71.8	87.8	78.4	80.9
Overall (n = 82,429)							
F	95.3	91.2	89.7	89.0	94.8	87.3	91.7
P	95.5	91.8	90.2	89.7	95.2	86.9	92.0
R	95.0	90.6	89.2	88.3	94.5	87.8	91.4

A separate evaluation is given for numbered, adjunct-like and external arguments, in addition to the overall evaluation. Note that all external arguments are included in the numbered and adjunct-like argument evaluations, seeing that each external argument is also one of these argument types. *N* is the total number of double annotated arguments (i.e. including both annotations but not gold standard arguments)

representation. Most importantly we add features extracted from morphological tags which were entirely absent in the English SemEval-14 data but are of obvious importance for Finnish SRL. We also incorporate features based on the predicted sense of the predicate. The thus modified system achieved a labeled F_1 -score of 76.60 %, which is a 2.8pp improvement to the initial Finnish SRL baseline of Kanerva and Ginter (2014). A small gain is also seen in the unlabeled F_1 -score, which is 90.43 % for this system. In comparison, the CoNLL-09 Shared Task results for the winning system of the SRL-only task, range from 75.99 % F_1 -score for German to 85.44 % F_1 score for English, for an average of 80.47 % F_1 -score across the seven tested languages (Hajič et al. 2009). Taking into account the differences in data sizes, among the systems, and in the exact annotation tasks, we can conclude that the 76.60 % obtained for Finnish is well within the expected range for an initial baseline.

10 Conclusions

This paper has presented the first semantic role labeling resource for general Finnish, the Finnish PropBank. This work builds on top of the previously existing Turku Dependency Treebank (Haverinen et al. 2013b), which consists of 204,399

tokens of written Finnish and uses the well-known Stanford Dependency scheme (de Marneffe and Manning 2008a, b; de Marneffe et al. 2013) as its syntax annotation scheme. The PropBank enables novel research in Finnish language technology, as this previously unavailable resource allows researchers to develop and test their systems for Finnish semantic role labeling.

The PropBank scheme (Palmer et al. 2005) used in this work is targeted at running text annotation of semantic roles, and it tackles the problem of defining semantic roles one verb at a time. Each verb is given a number of *framesets*, corresponding to coarse-grained senses, and each frameset is given a number of *arguments*. Arguments receive numbered labels, such as *arg0* or *arg3*, and free text descriptions.

The project had a total of six different annotators, who also acted as framers. A combination of double and single annotation was used in order to maximize the speed of the work while controlling annotation quality. According to our annotator accuracy measurements, which cover 43.0 % of all verb occurrences, the overall F₁-score of the PropBank was 91.7 %, indicating a high overall annotation quality. In addition, we found that numbered arguments were the easiest to annotate, while adjunct-like and external arguments were somewhat more difficult, which is in line with the results reported on English by Palmer et al. (2005).

The main contribution of the paper is the PropBank itself. This resource is available under a free license and at no cost at the address <http://bionlp.utu.fi/>. The data released contains both the framesets (all descriptions are both in Finnish and in English) and the annotated arguments on top of TDT.

In addition, this work also showed the compatibility of the SD and PropBank schemes, and the utility of the *conjunct propagation and additional dependencies* layer in TDT. Out of all numbered arguments and ArgMs (disregarding the verb *olla*), 81.3 % were direct syntactic dependents of the verbs when considering only the *base* layer of the treebank, but when considering both annotation layers, 93.2 % of all arguments and ArgMs were direct dependents. Due to the syntactic attachments in the SD scheme, the copular verb *olla* forms its own special case, which produces a considerable amount of external arguments: copular verbs cover 7.5 % of all verb occurrences of the corpus, and they produce a total of 10,932 external arguments.

As a result of a system where the same frameset can be assigned to several verbs simultaneously, the Finnish PropBank also contains a web, albeit incomplete, of links between verbs that have identical (or near-identical) argument structures. These links were used to provide frame information for rare verbs in the system of Kanerva and Ginter (2014).

We further established the baseline for Finnish automated SRL using two machine learning-based SRL methods, showing that the performance is roughly on a par with a number of other languages, as reported in the CoNLL-09 Shared Task. Further development and refinement of Finnish SRL systems constitutes a natural future work, building on the data. These systems can, in turn, be used to support further applications in for instance information extraction, machine translation or question answering. The PropBank itself can be extended with noun argument structures, resulting in a NomBank (Meyers et al. 2004b, a), and further, it can be modified according to the guidelines presented by Wanner et al. (2012) to support applications in text generation.

Acknowledgments This work has been supported by the Academy of Finland, the Emil Aaltonen Foundation and the University of Turku Foundation.

References

- (2009) *Collins english dictionary—30th anniversary edition*. HarperCollins Publishers.
- Baker, C.F., Fillmore, C.J., & Lowe, J.B. (1998). The Berkeley FrameNet project. In *Proceedings of COLING-ACL'98* (pp. 86–90).
- Barnickel, T., Weston, J., Collobert, R., Mewes, H. W., & Stümpfen, V. (2009). Large scale application of neural network based semantic role labeling for automated relation extraction from biomedical texts. *PLoS One*, 4(7). doi:[10.1371/journal.pone.0006393](https://doi.org/10.1371/journal.pone.0006393).
- Carreras, X., & Màrques, L. (2004). Introduction to the conll-2004 shared task: Semantic role labeling. In *CoNLL Proceedings of CoNLL-2004* (pp. 89–97).
- Carreras, X., & Màrquez, L. (2005). Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proceedings CoNLL-2005* (pp. 152–164).
- Choi, J., & Palmer, M. (2010). Retrieving correct semantic boundaries in dependency structure. In *Proceedings of LAW IV* (pp. 91–99).
- Comrie, B. (1985). Causative verb formation and other verb-deriving morphology. In T. Shopen (Ed.), *Language typology and syntactic description. Vol 3: Grammatical categories and the lexicon* (pp. 309–348). Cambridge: Cambridge University Press.
- Dang, H.T., Kipper, K., Palmer, M., & Rosenzweig, J. (1998). Investigating regular sense extensions based on intersective Levin classes. In *Proceedings of COLING-ACL'98* (pp. 293–299).
- de Marneffe, M.C., & Manning, C. (2008a). Stanford typed dependencies manual. In *Technical report*, Stanford University, revised for Stanford Parser v. 2.0.4 in November 2012.
- de Marneffe, M.C., & Manning, C. (2008b). Stanford typed dependencies representation. In *Proceedings of COLING'08, workshop on cross-framework and cross-domain parser evaluation* (pp. 1–8).
- de Marneffe, M.C., Connor, M., Silveira, N., Bowman, S.R., Dozat, T., & Manning, C.D. (2013). More constructions, more genres: Extending stanford dependencies. In *Proceedings of Depling'13* (pp. 187–196).
- Duran, M.S., & Aluísio, S.M. (2011). Propbank-br: A Brazilian treebank annotated with semantic role labels. In *Proceedings of STIL'11* (pp. 1862–1867).
- Gildea, D., & Jurafsky, D. (2002). Automatic labeling of semantic roles. *Computational Linguistics*, 28(3), 245–288.
- Giuglea, A.M., & Moschitti, A. (2004). Knowledge discovery using FrameNet, VerbNet and PropBank. In *Proceedings of the workshop on ontology and knowledge discovering at ECML, Pisa, Italy*.
- Giuglea, A. M., & Moschitti, A. (2006). Semantic role labeling via FrameNet, VerbNet and PropBank. In *Proceedings of the 21st international conference on computational linguistics and 44th annual meeting of the association for computational linguistics, association for computational linguistics, Sydney, Australia* (pp. 929–936).
- Hajič, J., Ciaramita, M., Johansson, R., Kawahara, D., Martí, M.A., Màrquez, L., Meyers, A., Nivre, J., Padó, S., Štěpánek, J., Straňák, P., Surdeanu, M., Xue, N., & Zhang, Y. (2009). The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of CoNLL'09* (pp. 1–18).
- Hakulinen, A., Vilkuna, M., Korhonen, R., Koivisto, V., Heinonen, T.R., & Alho, I. (2004). Iso suomen kielioppi / Grammar of Finnish. Suomalaisen kirjallisuuden seura.
- Haverinen, K. (2012). Syntax annotation guidelines for the Turku Dependency Treebank. Technical Report 1034, Turku Centre for Computer Science (2nd ed), revised for the treebank release of July 2013.
- Haverinen, K., Ginter, F., Laippala, V., Viljanen, T., & Salakoski, T. (2010). Dependency-based propbanking of clinical Finnish. In *Proceedings of LAW IV* (pp. 137–141).
- Haverinen, K., Laippala, V., Kohonen, S., Missilä, A., Nyblom, J., Ojala, S., Viljanen, T., Salakoski, T., & Ginter, F. (2013a). Towards a dependency-based PropBank of general Finnish. In *Proceedings of NODALIDA'13* (pp. 41–57).
- Haverinen, K., Nyblom, J., Viljanen, T., Laippala, V., Kohonen, S., Missilä, A., Ojala, S., Salakoski, T., & Ginter, F. (2013b). Building the essential resources for Finnish: The Turku Dependency Treebank. Language resources and evaluation In press. doi:[10.1007/s10579-013-9244-1](https://doi.org/10.1007/s10579-013-9244-1).

- Johansson, R., & Nugues, P. (2008). The effect of syntactic representation on semantic role labeling. In *Proceedings of Coling'08* (pp. 393–400).
- Kanerva, J., & Ginter, F. (2014). Post-hoc manipulations of vector space models with application to semantic role labeling. In *Proceedings of the 2nd workshop on continuous vector space models and their compositionality (CVSC)@ EACL* (pp. 1–10).
- Kanerva, J., Luotolahti, J., & Ginter, F. (2014). Turku: Broad-coverage semantic parsing with rich features. In *Proceedings of the international workshop on semantic evaluation, SemEval, to appear*.
- Levin, B., & Hovav, M. R. (1994). *Unaccusativity: At the syntax–lexical semantics interface*, *Linguistic Inquiry*, vol 26. Cambridge: MIT Press.
- Lindén, K., Silfverberg, M., & Pirinen, T. (2009). HFST tools for morphology—An efficient open-source package for construction of morphological analyzers. In C. Mahlow & M. Piotrowski (Eds.), *State of the art in computational morphology. communications in computer and information science* (vol. 41, pp. 28–47). Berlin: Springer.
- Marcus, M., Marcinkiewicz, M. A., & Santorini, B. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2), 313–330.
- Meyers, A., Reeves, R., Macleod, C., Szekely, R., Zielinska, V., Young, B., & Grishman, R. (2004a). Annotating noun argument structure for NomBank. In *Proceedings of LREC-2004*.
- Meyers, A., Reeves, R., Macleod, C., Szekely, R., Zielinska, V., Young, B., & Grishman, R. (2004b). The NomBank project: An interim report. In *Proceedings of the NAACL/HLT workshop on frontiers in corpus annotation*.
- Moschitti, A., Pighin, D., & Basili, R. (2008). Tree kernels for semantic role labeling. *Computational Linguistics*, 34(2), 194–224.
- Nyblom, J., Kohonen, S., Haverinen, K., Salakoski, T., & Ginter, F. (2013). Predicting conjunct propagation and other extended Stanford dependencies. In *Proceedings of the international conference on dependency linguistics (Depling 2013)* (pp. 252–261).
- Palmer, M., Gildea, D., & Kingsbury, P. (2005). The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1), 71–106.
- Palmer, M., Bhatt, R., Narasimhan, B., Rambow, O., Sharma, D.M., & Xia, F. (2009). Hindi syntax: Annotating dependency, lexical predicate–argument structure, and phrase structure. In *Proceedings of ICON'09*.
- Palmer, M., Gildea, D., & Xue, N. (2010). *Semantic role labeling*. San Rafael, California: Morgan & Claypool Publishers.
- Paulsen, G. (2011). *Causation and dominance*. PhD thesis, Åbo Akademi University, Finland.
- Perlmutter, D. (1978). Impersonal passives and the unaccusative hypothesis. In *Proceedings of the fourth annual meeting of the Berkeley Linguistic Society* (pp. 157–189).
- Pirinen, T. (2008). *Suomen kielen äärellistilainen automaattinen morfologinen jäsenen avoimen lähdekoodin resurssien*. Master's thesis, University of Helsinki.
- Pradhan, S. S., Ward, W., & Martin, J. H. (2008). Towards robust semantic role labeling. *Computational Linguistics*, 34(2), 289–310.
- Punyakanok, V., Roth, D., & Wt, Yih. (2008). The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2), 258–288.
- Surdeanu, M., Mărques, L., Carreras, X., & Comas, P. R. (2007). Combination strategies for semantic role labeling. *Computational Linguistics*, 29, 105–151.
- Surdeanu, M., Johansson, R., Meyers, A., Mărquez, L., & Nivre, J. (2008). The CoNLL 2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proceedings of CoNLL'08* (pp. 159–177).
- Toutanova, K., Haghighi, A., & Manning, C. D. (2008). A global joint model for semantic role labeling. *Computational Linguistics*, 34(2), 161–192.
- Wanner, L., Mille, S., & Bohnet, B. (2012). Towards a surface realization-oriented corpus annotation. In *Proceedings of INLG '12* (pp. 22–30).
- Xue, N., & Palmer, M. (2009). Adding semantic roles to the Chinese treebank. *Natural Language Engineering*, 15(Special issue 01), 143–172.
- Zaghoulani, W., Diab, M., Mansouri, A., Pradhan, S., & Palmer, M. (2010). The revised Arabic PropBank. In *Proceedings of LAW IV* (pp. 222–226).